

## Fundamentals of Bicep Training

### Unit 1

- **What is infrastructure as code?**
- **Defining Infrastructure as code**
- **Why use Infrastructure as code?**
  
- **Increase confidence**
  - Integration with current processes
  - Consistency
  - Automated scanning
  - Secret management
  - Access control
  - Avoid configuration drift
  
- **Manage multiple environments**
  - Provision new environments
  - Non-production environments
  - Disaster recovery
  
- **Better understand your cloud resources**
  - Audit trail
  - Documentation
  - Unified system
  - Better understanding of cloud infrastructure

### Imperative and declarative code

### Unit 2

### What is Azure Resource Manager?

### Azure Resource Manager concepts

### Terminology

- Resource
- Resource group
- Subscription
- Management group:
- Azure Resource Manager template

Benefits

**Operations: Control plane and data plane**

**Why use ARM templates?**

- Repeatable results
- Orchestration
- Preview
- Testing and Validation
- Modularity
- CI/CD integration
- Extensibility

**JSON and Bicep templates**

Unit 3

**What is Bicep?**

**Benefits of Bicep**

- Simpler syntax:
- Modules
- Automatic dependency management
- Type validation and IntelliSense

Unit 4

**How Bicep works**

**Bicep deployment**

**Comparing JSON and Bicep**

## Unit 5

### When to use Bicep

#### Is Bicep the right tool?

##### When is Bicep the right tool?

- Azure-native
- Azure integration
- Azure support
- No state management
- Easy transition from JSON

##### When is Bicep not the right tool?

- Existing tool set:
- Multicloud
  
- **What is the main goal?**
- After completing this module, you'll be able to determine whether infrastructure as code is the right approach and tool for your organization.